



TUGAS AKHIR

**IMPLEMENTASI *ARTIFICIAL INTELLIGENCE* PADA
PERMAINAN *SNAKE* DENGAN
METODE *REINFORCEMENT LEARNING***

Sulistyawan Abdillah Rosyid
NIM. 11181079

Nisa Rizqiya Fadhliana, S.Kom., M.T.
Rizky Amelia, S.Si., M.Han.

Program Studi Informatika
Jurusan Matematika dan Teknologi Informasi
Institut Teknologi Kalimantan
Balikpapan, 2025

LEMBAR PERSETUJUAN

Proposal Tugas Akhir dengan judul:

**“IMPLEMENTASI *ARTIFICIAL INTELLIGENCE* PADA PERMAINAN
SNAKE DENGAN METODE *REINFORCEMENT LEARNING*”**

Yang disusun oleh:



Sulistyawan Abdillah Rosyid

NIM. 11181079

Telah diperiksa dan disetujui oleh dosen pembimbing:

Dosen Pembimbing Utama

Dosen Pembimbing Pendamping

Nisa Rizqiya Fadhliana, S.Kom., M.T.

NIP/NIPH. 198804102019032020

Rizky Amelia, S.Si., M.Han.

NIP/NIPH. 199308092022032016

KATA PENGANTAR

Puji syukur kepada Tuhan Yang Maha Esa atas berkat dan anugerah-Nya sehingga kami dapat menyelesaikan proposal tugas akhir yang berjudul:

“IMPLEMENTASI *ARTIFICIAL INTELLIGENCE* PADA PERMAINAN *SNAKE* DENGAN METODE *REINFORCEMENT LEARNING*”

Proposal tugas akhir ini merupakan salah satu syarat yang harus ditempuh untuk menyelesaikan Program Sarjana di Program Studi Informatika, Jurusan Matematika Teknologi dan Informasi, Institut Teknologi Kalimantan (ITK) Balikpapan. Untuk itu penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Tuhan YME atas segala nikmat yang diberikan sehingga dapat terlaksana pembuatan laporan ini.
2. Ibu Nisa Rizqiya Fadhliana, S.Kom., M.T. selaku Dosen Pembimbing Utama dan Koordinator Program Studi Informatika Jurusan Matematika Teknologi dan Informasi ITK.
3. Ibu Rizky Amelia, S.Si., M.Han. selaku Dosen Pembimbing Pendamping dan Dosen Wali dari penulis.
4. Bapak Ibu Seluruh Dosen serta Tenaga Kependidikan Program Studi Informatika Jurusan Matematika dan Teknologi Informasi ITK.
5. Khoirun Nisa Al Fahmi selaku istri dari penulis dan keluarga dari penulis yang selalu suportif untuk menyelesaikan laporan ini.
6. Serta semua pihak yang terlibat dalam penyusunan proposal tugas akhir ini.

Penulis menyadari bahwa penyusunan laporan Tugas Akhir ini masih jauh dari sempurna, karena itu kami mengharapkan segala kritik dan saran yang membangun. Semoga Tugas Akhir ini dapat bermanfaat bagi kita semua. Atas perhatiannya penulis ucapkan terima kasih.

Balikpapan, 10 September 2024

Penyusun

ABSTRAK

Penelitian ini membahas implementasi *Artificial Intelligence* (AI) pada permainan *Snake* menggunakan metode *Reinforcement Learning* (RL). Tujuan utama dari penelitian ini adalah mengembangkan agen yang mampu bermain *Snake* secara optimal melalui metode RL, dengan *Deep Q-Network* (DQN). Lingkungan simulasi *Snake* dikembangkan untuk melatih agen AI dengan fokus pada interaksi agen dengan lingkungan, seperti menghindari dinding dan memakan makanan untuk memaksimalkan nilai *reward*. Analisis dilakukan untuk mengevaluasi kinerja agen berdasarkan parameter-parameter yang mempengaruhi, termasuk struktur *reward*, ukuran *grid*, dan kecepatan permainan. Hasil penelitian menunjukkan bahwa RL yang diterapkan mampu meningkatkan performa agen dalam memainkan *Snake*, dengan evaluasi performa yang dibandingkan dengan metode lain. Penelitian ini berkontribusi pada pengembangan RL yang lebih efisien dan memiliki potensi aplikasi lebih luas dalam robotika, navigasi otonom, dan sistem kompleks lainnya.

Kata Kunci: Agent, Artificial Intelligence, Deep Q-Network, Reinforcement Learning

ABSTRACT

This study explores the implementation of Artificial Intelligence (AI) in the Snake game using the Reinforcement Learning (RL) method. The primary objective of the research is to develop an agent capable of playing Snake optimally through RL, specifically utilizing the Deep Q-Network (DQN) approach. A custom simulation environment was developed to train the AI agent, focusing on agent–environment interactions, such as avoiding collisions with walls and consuming food to maximize reward values. The performance of the agent was evaluated based on several influential parameters, including reward structure, grid size, and game speed. The results demonstrate that the applied Reinforcement Learning effectively improves the agent’s performance in playing Snake, with comparative analysis conducted against alternative methods. This research contributes to the development of more efficient Reinforcement Learning applications and highlights its potential for broader use in robotics, autonomous navigation, and other complex systems.

Keywords: Agent, Artificial Intelligence, Deep Q Network, Reinforcement Learning

DAFTAR ISI

LEMBAR PERSETUJUAN.....	ii
KATA PENGANTAR	iii
ABSTRAK	v
ABSTRACT	vi
DAFTAR ISI.....	vii
DAFTAR GAMBAR	ix
DAFTAR TABEL.....	x
1. BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	3
1.3 Tujuan Penelitian	3
1.4 Batasan Penelitian	3
1.5 Manfaat Penelitian	4
1.6 Kerangka Pemikiran.....	5
2. BAB 2 TINJAUAN PUSTAKA	8
2.1 <i>Artificial intelligence (AI)</i>	8
2.2 <i>Machine Learning</i>	9
2.3 <i>Reinforcement Learning</i>	10
2.3.1. Q-Learning	12
2.4 Teori Permainan	14
2.5 <i>Deep Learning</i>	15

2. 5. 1.	<i>Deep-Q-Network (DQN)</i>	16
2. 5. 2.	Experience Replay	17
2. 5. 3.	Policy Gradient Method	18
2.6	Penelitian Terdahulu	19
3.	BAB 3 METODOLOGI PENELITIAN.....	26
3.1	Gambaran Besar Penelitian	26
3.2	Diagram Alir Penelitian	27
3.3	Prosedur Penelitian.....	28
3.3.1	Studi Literatur	28
3.3.2	Pengembangan Lingkungan Simulasi	28
3.3.3	Penerapan Algoritma <i>Reinforcement Learning</i>	29
3.3.4	Menganalisis Pengujian dan Validasi Hasil.....	32
3.4	Jadwal Penelitian.....	33
	DAFTAR PUSTAKA	35

DAFTAR GAMBAR

Gambar 1. 1 Fishbone Diagram	6
Gambar 2. 1 Skema <i>Reinforcement Learning</i>	11
Gambar 3. 1 Diagram Alir Penelitian	27
Gambar 3. 2 Skema <i>Deep Q Network</i>	30

DAFTAR TABEL

Tabel 2.1 Penelitian Terdahulu	19
Tabel 3.1 Rencana Penelitian.....	33

BAB 1

PENDAHULUAN

Penelitian ini berfokus pada penerapan *Artificial Intelligence* (AI) dalam permainan Snake dengan menggunakan metode *Reinforcement Learning*. Permainan Snake dipilih karena strukturnya yang sederhana namun cukup kompleks untuk dijadikan lingkungan pembelajaran bagi agen cerdas. Penelitian ini bertujuan untuk menguji efektivitas metode tersebut dalam mengembangkan agen yang adaptif, serta memberikan gambaran mengenai potensi penerapan AI pada gim klasik sebagai media pembelajaran dan eksperimen algoritma pembelajaran mesin.

Pada bab ini menjelaskan mengenai pendahuluan dari penelitian yang terdiri dari latar belakang, rumusan masalah, tujuan, manfaat dari dilakukannya penelitian ini.

1.1 Latar Belakang

Artificial intelligence (AI) merupakan sebuah istilah yang sering digaungkan akhir-akhir ini. Salah satu penyebabnya adalah saat ChatGPT dirilis oleh OpenAI pada November 2022. Sejak saat itu banyak produk-produk AI lain mulai bermunculan. Metode *Deep Learning* (DL), *Natural Language Processing* (NLP), *Computer Vision* banyak digunakan oleh perusahaan penghasil AI. AI sendiri sebenarnya bukanlah sesuatu yang baru. Gagasan mengenai AI ini ditemukan pada tahun 1956, tetapi karena keterbatasan kala itu membuat gagasan AI ini menjadi sesuatu yang diremehkan oleh sebagian besar orang. Saat ini AI dibuat dengan tiga cara yaitu *Supervised Learning*, *Unsupervised Learning*, dan *Reinforcement Learning*.

Perkembangan teknologi komputer dan *Artificial Intelligence* (AI) telah mengalami kemajuan yang pesat dalam beberapa tahun terakhir. Salah satu cabang AI yang mendapatkan perhatian khusus adalah cabang *Machine Learning* (ML). Cabang ini mencakup beragam metode dan algoritma yang mengajarkan komputer untuk cara untuk mengambil keputusan dan tindakan. Dari beragam model *Machine Learning*,

Reinforcement Learning (RL) cukup menonjol sebagai teknik yang menjanjikan untuk aplikasi di lingkungan yang dinamis dan kompleks. Dalam model ini sebuah agen belajar melalui interaksinya dengan lingkungan untuk memaksimalkan sebuah *reward*.

Permainan “*Snake*” adalah sebuah gim klasik yang ideal untuk mengembangkan algoritma *Reinforcement Learning* (RL) karena kesederhanaan dan tantangan pada gim ini. Dalam gim ini, agen (*Snake*) diharuskan untuk mencari dan mendapatkan makanan yang muncul secara acak sembari menghindari agar tidak tertabrak dinding atau diri agen sendiri yang selalu bertumbuh setiap memakan sebuah makanan. Penerapan AI pada gim *Snake* menggunakan teknik RL memberikan beberapa kemungkinan, termasuk sebagai platform yang dapat diskalakan untuk menguji algoritma *Reinforcement Learning* dalam lingkungan yang terkendali dan dinamis. Aplikasi ini juga dapat disesuaikan dengan bidang lainnya seperti robotika dan navigasi. Gim ini memungkinkan proses pembelajaran dan adaptasi agen AI divisualisasikan dengan jelas. Hal ini memberikan wawasan penting tentang bagaimana algoritma *Reinforcement Learning* belajar dari pengalaman.

Pada jurnal sebelumnya milik Crespo dan Wichert (2020), dalam pembelajaran penguatan *Reinforcement Learning*, *Deep Q-Networks* (*DQN*) digunakan secara luas untuk mempelajari keputusan optimal dalam lingkungan permainan, seperti pada gim Atari. *DQN* memanfaatkan jaringan saraf untuk mendekati fungsi nilai yang memprediksi seberapa baik suatu tindakan dalam situasi tertentu. Peningkatan penting pada metode *DQN* meliputi *Averaged-DQN*, yang menstabilkan proses pembelajaran dengan mengurangi variabilitas dalam perkiraan *Q-values*, serta *Rainbow DQN*, yang menggabungkan beberapa perbaikan, termasuk *Double Deep Q-Learning*, *dueling networks*, dan *prioritized replay* untuk meningkatkan performa algoritma dalam berbagai tugas gim (Crespo dan Wichert, 2020).

Penelitian ini bertujuan untuk mengembangkan agen yang mampu bermain *Snake* dengan sangat baik dan mengevaluasi efisiensi dan efektivitas berbagai algoritma RL yang diterapkan untuk permainan *Snake*. Selain itu, penelitian ini juga bertujuan untuk menemukan masalah dan peluang dalam penerapan RL dalam permainan *Snake* dan memberikan saran untuk penelitian tambahan. Penelitian ini diharapkan dapat

meningkatkan pemahaman kita tentang *Reinforcement Learning* secara teoritis dan dalam aplikasinya yang kompleks dan dinamis. Ini juga akan membuka jalan untuk inovasi lebih lanjut dalam penggunaan AI untuk aplikasi yang kompleks dan dinamis.

1.2 Perumusan Masalah

Berdasarkan latar belakang, maka didapatkan rumusan masalah sebagai berikut:

1. Bagaimana mengimplementasikan *Reinforcement Learning* untuk membuat sebuah agen AI yang mampu bermain gim “*Snake*” secara baik ?
2. Apa saja parameter yang memengaruhi kinerja agen dalam permainan dan bagaimana pengaruhnya ?
3. Bagaimana mengukur efektivitas *Reinforcement Learning* dengan metode Bellman dibandingkan dengan metode lain dalam permainan Snake?

1.3 Tujuan Penelitian

Sesuai dengan rumusan masalah, tujuan dari penelitian ini adalah sebagai berikut:

1. Mengimplementasikan *Reinforcement Learning* pada sebuah agen untuk gim “*Snake*” secara baik.
2. Mengetahui parameter yang memengaruhi kinerja agen dalam permainan dan bagaimana pengaruh dari parameter tersebut terhadap kinerja agen.
3. Membandingkan efektivitas dari *Reinforcement Learning* pada gim “*Snake*” dengan algoritma *Bellman* dan *Monte Carlo*.

1.4 Batasan Penelitian

Adapun batasan masalah pada penelitian adalah sebagai berikut:

1. Algoritma yang digunakan adalah variasi dari jenis-jenis tertentu dari *Reinforcement Learning Deep Q-Network* atau varian lain yang relevan.

2. Implementasi akan dilakukan dengan *framework* atau *library* tertentu seperti PyTorch.
3. Pelatihan agen akan difokuskan pada permainan *Snake* 2 dimensi tanpa modifikasi aturan atau lingkungan permainan yang kompleks.
4. Analisis berfokus pada bagaimana variasi parameter ini mempengaruhi kinerja agen dalam hal skor rata-rata, panjang antrean, dan waktu bertahan hidup.
5. Parameter yang akan dianalisis meliputi ukuran *grid*, kecepatan permainan, struktur *reward system*, dan parameter internal *Reinforcement Learning* seperti *learning rate*, *discount factor*, dan *exploration rate*.
6. Kinerja agen yang diperoleh dari *Reinforcement Learning* dilakukan dengan pendekatan algoritma *Bellman* dan *Monte Carlo*.
7. Evaluasi didasarkan pada metrik kinerja seperti skor rata-rata yang dicapai oleh agen, panjang antrean maksimum yang dicapai, dan stabilitas kinerja.
8. Penelitian ini dibatasi pada evaluasi dalam lingkungan simulasi, tanpa penerapan langsung di luar lingkup permainan *Snake*.

1.5 Manfaat Penelitian

Adapun manfaat secara umum dari penelitian ini adalah memberikan wawasan lebih dalam tentang cara kerja metode *Reinforcement Learning* dalam lingkungan dinamis berbasis aturan seperti permainan *Snake*. Hasil penelitian ini dapat mengarah pada pengembangan dan peningkatan *Reinforcement Learning* yang lebih efisien dan efektif. Permainan Ular berfungsi sebagai studi kasus yang sangat baik untuk membuat prototipe aplikasi RL sebelum menerapkannya pada masalah dunia nyata yang lebih kompleks.

Peneliti yang berpartisipasi dalam penelitian ini akan mendapatkan pengalaman langsung menerapkan dan menguji algoritma AI serta mengembangkan keterampilan analisis data dan pemecahan masalah. Studi ini memungkinkan peneliti untuk mengembangkan keterampilan teknis dalam pemrograman, menggunakan *library* AI PyTorch, dan menerapkan metode ilmiah. Selain itu, teknik dan wawasan yang

diperoleh dengan menerapkan *Reinforcement Learning* pada gim *Snake* dapat diterapkan pada berbagai bidang lainnya, seperti robotika, navigasi otonom, dan pengelolaan sumber daya.

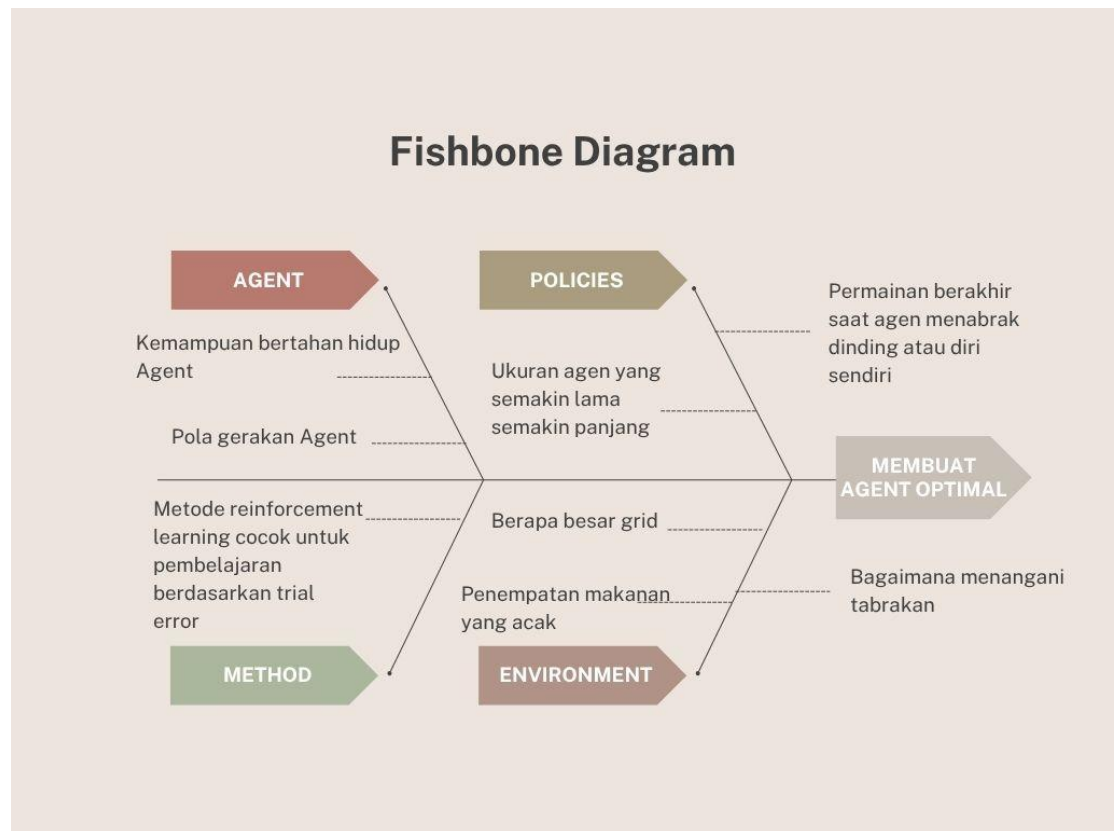
Penelitian ini juga memiliki manfaat signifikan bagi industri permainan. Dengan memahami dan menerapkan RL yang lebih baik, pengembang permainan dapat menciptakan pengalaman bermain yang lebih menarik dan adaptif. Algoritma yang lebih efisien dapat digunakan untuk mengembangkan *non player character*(NPC) yang lebih cerdas, meningkatkan interaksi pemain, dan menciptakan tantangan yang lebih dinamis dalam permainan. Selain itu, teknik yang diperoleh dari penelitian ini dapat membantu dalam pengembangan sistem rekomendasi yang lebih baik, yang dapat meningkatkan keterlibatan pemain dan retensi pengguna dalam permainan.

1.6 Kerangka Pemikiran

Dalam upaya mengembangkan agen kecerdasan buatan yang mampu bermain permainan Snake secara optimal, diperlukan suatu kerangka pemikiran yang sistematis. Permainan Snake dipilih karena meskipun tampak sederhana, ia menawarkan kompleksitas yang cukup tinggi dalam hal pengambilan keputusan secara real-time, adaptasi terhadap kondisi lingkungan, serta evaluasi strategi berdasarkan reward. Oleh karena itu, metode *Reinforcement Learning* (RL), khususnya pendekatan *Deep Q-Network* (DQN), dianggap paling tepat karena memungkinkan agen belajar dari pengalaman melalui proses trial-and-error untuk meningkatkan kinerja dari waktu ke waktu.

Kerangka pemikiran dalam penelitian ini berfokus pada hubungan antara berbagai komponen kunci yang memengaruhi kinerja agen AI, seperti algoritma yang digunakan, konfigurasi lingkungan permainan, parameter pelatihan, dan evaluasi performa. Fishbone diagram yang digunakan memvisualisasikan bagaimana faktor-faktor seperti kemampuan agen, aturan permainan, metode pembelajaran, dan kondisi simulasi bekerja secara sinergis untuk mencapai tujuan akhir, yaitu menciptakan agen

AI yang tidak hanya mampu bertahan hidup dalam permainan, tetapi juga mampu memaksimalkan *reward* secara konsisten. Dengan pendekatan ini, penelitian diharapkan mampu memberikan gambaran menyeluruh terhadap proses pengembangan agen AI berbasis RL dalam lingkungan permainan yang dinamis dan penuh tantangan



Gambar 1. 1 Fishbone Diagram

Fishbone yang digambarkan oleh gambar 1.1 memiliki berbagai faktor yang memengaruhi keberhasilan dalam menciptakan agen kecerdasan buatan (AI) yang optimal untuk permainan *Snake* menggunakan pendekatan *Reinforcement Learning*. Faktor pertama berasal dari sisi *Agent*, di mana kemampuan agen untuk bertahan hidup serta pola gerakannya menjadi aspek krusial dalam menentukan seberapa baik agen tersebut dapat merespons perubahan lingkungan dalam permainan. Agen yang

memiliki strategi gerakan adaptif cenderung memiliki peluang lebih besar untuk bertahan lama dan memperoleh skor lebih tinggi.

Selanjutnya, dari aspek *Policies*, terdapat aturan permainan yang harus diikuti agen, seperti bertambahnya panjang tubuh ular seiring waktu serta kondisi kekalahan ketika agen menabrak dinding atau tubuhnya sendiri. Aturan-aturan ini menambah kompleksitas tantangan yang harus dihadapi oleh agen dalam setiap langkah pengambilan keputusan.

Kemudian dari sisi *Method*, *Reinforcement Learning* dianggap metode yang tepat karena sesuai untuk skenario “*trial and error*”, memungkinkan agen belajar dari setiap kegagalan dan perolehan reward secara langsung selama proses bermain.

Faktor *Environment* juga memberikan pengaruh signifikan terhadap kinerja agen. Permainan Snake menyajikan lingkungan yang berubah-ubah, misalnya ukuran grid yang bervariasi dan penempatan makanan yang dilakukan secara acak. Hal ini mengharuskan agen untuk mampu menyesuaikan strategi dengan kondisi yang tidak sepenuhnya dapat diprediksi. Lingkungan tersebut juga menuntut adanya mekanisme yang efisien dalam menangani situasi benturan, baik dengan dinding maupun dengan tubuh agen sendiri. Semua faktor ini saling berkaitan dan membentuk sistem pembelajaran yang kompleks namun terukur, dengan tujuan akhir untuk menciptakan agen AI yang cerdas, adaptif, dan optimal dalam menjalankan tugasnya di dalam permainan Snake

..

BAB 2

TINJAUAN PUSTAKA

Pada bab ini akan menjelaskan terkait tinjauan pustaka yang digunakan pada penelitian.

2.1 *Artificial intelligence (AI)*

Artificial intelligence (AI) adalah cabang ilmu komputer yang berfokus pada pengembangan sistem yang mampu melakukan tugas yang biasanya membutuhkan kecerdasan manusia, seperti pengenalan suara, pengambilan keputusan, dan terjemahan bahasa, AI didefinisikan sebagai studi tentang agen cerdas, yaitu entitas yang dapat merasakan lingkungan mereka dan mengambil tindakan untuk memaksimalkan peluang keberhasilan dalam mencapai tujuan tertentu. Dalam definisi ini, AI mencakup berbagai pendekatan, mulai dari logika simbolik hingga pembelajaran mesin yang lebih modern (Russell, 2020).

Dalam konteks pembelajaran mesin, melibatkan algoritma yang memungkinkan komputer belajar dari data dan memperbaiki performa mereka dari waktu ke waktu tanpa diprogram secara eksplisit. Pendekatan ini telah berkembang pesat dalam beberapa tahun terakhir dan diterapkan dalam berbagai bidang, termasuk kesehatan, transportasi, dan keuangan, untuk membuat prediksi yang lebih akurat dan otomatisasi proses yang kompleks (Goodfellow, dkk., 2016).

AI dapat dikelompokkan menjadi beberapa kategori utama berdasarkan kemampuannya, yaitu AI lemah (*narrow AI*) dan AI kuat (*general AI*). AI lemah dirancang untuk melakukan tugas tertentu dengan sangat baik, seperti pengenalan wajah atau pengolahan bahasa alami, tetapi tidak memiliki kemampuan untuk melakukan tugas di luar dari spesialisasinya. Sementara itu, AI kuat adalah sebuah konsep di mana mesin memiliki kecerdasan umum yang setara dengan manusia, dengan kemampuan untuk belajar dan beradaptasi dalam berbagai situasi. Meskipun

AI kuat masih merupakan tujuan jangka panjang yang belum tercapai, AI lemah sudah diterapkan luas dalam kehidupan sehari-hari (Tegmark, 2017).

Perkembangan AI yang pesat membawa potensi besar tetapi juga tantangan serius terkait etika, keamanan, dan dampak sosial. Bostrom mengemukakan bahwa ketika AI mencapai tingkat super intelligence, atau kecerdasan yang jauh melampaui kecerdasan manusia, ini dapat membawa perubahan yang sangat besar dalam peradaban manusia, termasuk peluang dan ancaman eksistensial. Oleh karena itu, Bostrom menekankan pentingnya penelitian untuk memastikan bahwa AI berkembang dengan cara yang aman dan bermanfaat bagi umat manusia (Bostrom, 2014).

2.2 *Machine Learning*

Belakangan ini Pembelajaran Mesin / *Machine Learning* sungguh banyak digunakan, bahkan bisa lebih banyak dari yang kita perkirakan. Contohnya saat kita mencari “Hadiah” untuk teman atau sahabat kita di internet. Kita akan mengetikkan sebuah kata kunci maka muncullah hal-hal yang berkaitan dengan kata kunci tersebut. Saat kita membuka kotak email kita lagi, tanpa disadari, kita akan menemukan sebuah email sejenis masuk ke kotak email kita atau mungkin dapat kita temukan pada *spam*. Selanjutnya saat pergi ke sebuah supermarket untuk membeli popok untuk dijadikan hadiah kepada sahabat. Saat ingin membayar, kasir memberikan sebuah kupon diskon untuk bir. Aksi kasir memberikan sebuah kupon diskon pada orang yang membeli popok ditunjukkan oleh *software* juga akan membeli bir. Data itulah yang dapat digunakan untuk mencari strategi untuk bisnis.

Apa yang akan kita cari pada sebuah data mentah itu tidak akan jelas. Contohnya dalam mendeteksi sebuah email *spam* mencari sebuah kata dalam email tentu tidak akan terlalu membantu. Bahkan akan terjadi kesalahan-kesalahan pendeteksian *email spam*. Karenanya akan lebih baik jika mencari gabungan dari beberapa kata yang ada dalam sebuah *email*, digabungkan dengan panjang *email*, laporan *spam* dari user lain, dan banyak hal lainnya. Dengan pertimbangan itu semua

akan menjadi jelas sebuah *email* adalah *spam* atau tidak. Inilah salah satu kegunaan *Machine Learning* untuk mengubah data mentah menjadi informasi yang dapat digunakan (Harrington, 2012).

Machine Learning adalah cabang dari kecerdasan buatan (*artificial intelligence*) yang berfokus pada pengembangan algoritma dan teknik yang memungkinkan komputer untuk belajar dari data. Dalam pengertian ini, mesin dapat belajar dari pengalaman atau data masa lalu untuk membuat prediksi atau keputusan tanpa harus diprogram secara eksplisit untuk setiap tugas (Mitchell, 1997).

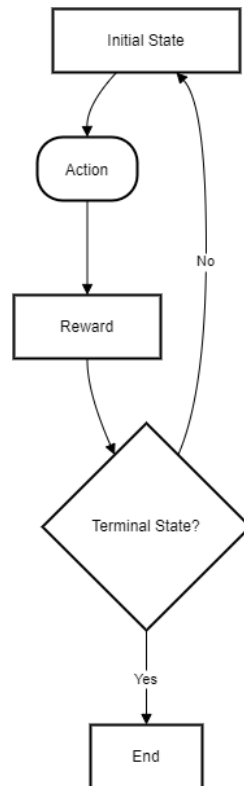
Machine Learning melibatkan proses di mana model atau algoritma "dilatih" dengan menggunakan data historis. Setelah proses pelatihan ini, model dapat digunakan untuk membuat prediksi atau menyelesaikan masalah tertentu berdasarkan data baru. Proses ini mencakup berbagai teknik, seperti *supervised learning*, *unsupervised learning*, dan *Reinforcement Learning*, yang masing-masing memiliki aplikasi dan pendekatan yang berbeda tergantung pada jenis data dan tujuan analisis (Alpaydin, 2020).

Machine Learning telah menjadi landasan utama dalam berbagai bidang, mulai dari pengenalan suara dan gambar hingga analisis prediktif dalam bisnis dan industri. *Machine Learning* memungkinkan sistem untuk mengenali pola kompleks dalam data dan secara otomatis memperbaiki kinerjanya seiring waktu, membuatnya menjadi alat yang sangat kuat dalam era data besar saat ini (Goodfellow, dkk., 2016).

2.3 *Reinforcement Learning*

Reinforcement Learning (RL) adalah suatu metode pembelajaran mesin di mana sebuah agen belajar untuk membuat keputusan dengan melakukan aksi-aksi di dalam lingkungan tertentu guna memaksimalkan suatu nilai *reward* kumulatif. *Reinforcement Learning* menekankan pada interaksi antara agen dan lingkungan melalui percobaan dan kesalahan, di mana agen tersebut mengamati status lingkungannya, memilih aksi, dan kemudian menerima *feedback* berupa *reward*.

Proses ini memungkinkan agen untuk belajar secara mandiri tanpa memerlukan data pelatihan yang lengkap seperti dalam *supervised learning* (Sutton dan Barto, 2018).



Gambar 2. 1 Skema *Reinforcement Learning*

Dalam konteks ini, agen bertujuan untuk menemukan kebijakan (*policy*) yang menentukan bagaimana agen harus bertindak di berbagai situasi agar mendapatkan *reward* maksimal. *Reinforcement Learning* sering kali digunakan dalam skenario di mana keputusan harus dibuat secara berurutan dan dampaknya mungkin tidak segera terlihat, seperti dalam permainan video gim atau pengendalian robotika (Mnih, dkk., 2015).

Selain itu, *Reinforcement Learning* memiliki berbagai aplikasi di dunia nyata yang semakin berkembang. *Reinforcement Learning* telah digunakan untuk

mengalahkan juara dunia dalam permainan Go, melalui sistem bernama AlphaGo. Sistem ini memanfaatkan teknik *Monte Carlo Tree Search* (MCTS) bersama dengan *deep neural networks* untuk mengevaluasi posisi dan memilih langkah terbaik. Keberhasilan ini menyoroti potensi RL dalam menangani masalah kompleks yang membutuhkan pengambilan keputusan strategis di berbagai domain, mulai dari permainan hingga industri otomotif dan keuangan (Silver, dkk., 2016).

Penggunaan *Reinforcement Learning* juga mulai terlihat dalam pengoptimalan proses bisnis dan industri. Pada sektor energi, *Reinforcement Learning* digunakan untuk mengoptimalkan pengelolaan jaringan listrik pintar (*smart grid*) dan pengaturan sumber daya energi terbarukan. Agen *Reinforcement Learning* dapat dilatih untuk merespon perubahan permintaan dan penawaran listrik, serta memprediksi konsumsi energi untuk meningkatkan efisiensi dan stabilitas sistem (François-Lavet, dkk., 2018).

Secara umum, *Reinforcement Learning* terus berkembang seiring dengan peningkatan kemampuan komputasi dan kemajuan algoritma pembelajaran. Tantangan utama dalam *Reinforcement Learning* meliputi eksplorasi yang efisien, generalisasi terhadap situasi baru, dan pengurangan kompleksitas komputasi. Namun, dengan perkembangan teknik seperti *Deep Reinforcement Learning*, multi-agent *Reinforcement Learning*, dan *Meta Reinforcement Learning*, pendekatan ini memiliki potensi yang besar untuk diterapkan dalam berbagai disiplin ilmu dan industri (Li, 2018).

2.3.1. Q-Learning

Q-Learning bertujuan untuk menemukan kebijakan optimal yang memaksimalkan total *reward* jangka panjang. Pada dasarnya, *Q-Learning* menggunakan fungsi nilai aksi yang dikenal sebagai fungsi Q untuk mengestimasi seberapa baik tindakan tertentu yang dilakukan dalam suatu keadaan tertentu (Watkins, 1992).

Fungsi Q didefinisikan sebagai:

$$Q(s, a) = Q(s, a) + \alpha(r + \gamma \max_{a'} Q(s', a') - Q(s, a)) \dots \dots \dots (2.1)$$

Keterangan:

$Q(s, a)$: Nilai Q
s	: State / Keadaan
a	: Aksi
α	: <i>Learning Rate</i> / laju pembelajaran
r	: <i>Reward</i>
γ	: <i>Discount factor</i>
s'	: <i>State</i> Selanjutnya
a'	: Aksi selanjutnya
$\max_{a'} Q(s', a')$: Estimasi nilai maksimum Q untuk semua aksi (a') pada keadaan (s')

Nilai fungsi Q yang mewakili estimasi nilai dari suatu aksi a yang diambil pada keadaan s . *Learning rate* (laju pembelajaran), yaitu seberapa besar kita memperbarui nilai Q berdasarkan informasi baru. Nilai α biasanya berkisar antara 0 dan 1, di mana nilai yang lebih besar berarti pembelajaran lebih cepat, tetapi mungkin menyebabkan ketidakstabilan jika terlalu besar. *Reward* (hadiah) yang diterima setelah melakukan aksi a di keadaan s , dan pindah ke keadaan berikutnya s' . Ini adalah hadiah langsung yang diperoleh dari lingkungan. *Discount factor* (faktor diskon) menentukan seberapa besar pengaruh *reward* masa depan terhadap nilai Q saat ini. Nilai γ juga berkisar antara 0 dan 1, di mana 1 berarti *reward* masa depan dianggap sepenuhnya, dan 0 berarti hanya *reward* saat ini yang diperhitungkan. Next state (keadaan berikutnya) yang dicapai setelah mengambil aksi a dari keadaan s . Estimasi nilai maksimum Q untuk semua aksi a' yang bisa diambil pada keadaan berikutnya s' . Maksudnya, dari semua aksi yang mungkin, kita memilih aksi terbaik yang memaksimalkan nilai Q di keadaan s' (Sutton & Barto, 2018).

2.4 Teori Permainan

Teori permainan adalah cabang matematika yang mempelajari interaksi strategis antara agen rasional, di mana hasil yang dicapai tergantung pada keputusan yang diambil oleh semua pemain. Teori ini umumnya diterapkan pada situasi kompetitif dan kooperatif yang melibatkan pilihan strategis dari berbagai pemain. Dalam konteks permainan *Snake*, kita dapat menggunakan beberapa prinsip dasar dari teori permainan untuk menganalisis pola perilaku dan strategi optimal yang memungkinkan pemain untuk mencapai hasil terbaik.

Permainan *Snake* adalah permainan pemain tunggal di mana pemain mengendalikan ular yang bergerak di dalam arena untuk mengambil / memakan buah yang muncul di layar. Setiap kali ular memakan buah, panjang tubuhnya bertambah, sehingga membuat permainan lebih sulit karena pergerakan menjadi lebih terbatas. Pemain akan kalah jika ular menabrak dirinya sendiri atau dinding.

Dari perspektif teori permainan, *Snake* dapat dianggap sebagai permainan sekuensial deterministik, di mana setiap langkah pemain berpengaruh terhadap posisi masa depan ular, mengingat panjang tubuh ular yang bertambah dari waktu ke waktu. Dalam teori permainan, ini mirip dengan extensive form games, di mana pemain membuat keputusan secara bertahap dengan informasi lengkap tentang kondisi permainan pada setiap langkah.

Pada permainan *Snake*, pendekatan umum yang sering digunakan pemain adalah strategi serakah (*greedy strategy*), di mana pemain berusaha segera menuju buah yang tersedia dalam langkah tersingkat. Meskipun strategi ini efisien untuk tahap awal, permainan menjadi lebih kompleks seiring dengan bertambahnya panjang ular. Dalam teori permainan, strategi serakah tidak selalu optimal karena bisa menyebabkan kebuntuan di masa depan. Permainan *Snake* juga dapat dilihat sebagai permainan yang mengandalkan heuristik, yaitu aturan-aturan yang digunakan pemain untuk membuat keputusan cepat. Contoh heuristik dalam *Snake* adalah selalu menjaga agar ular bergerak dalam pola melingkar atau persegi agar tetap memiliki ruang gerak yang cukup. Dalam hal ini, teori permainan berbicara tentang bagaimana heuristik

membantu dalam membuat keputusan optimal di bawah tekanan waktu dan kondisi yang terbatas.

Dari perspektif ilmu komputer dan teori permainan, strategi optimal dalam *Snake* sering kali dikembangkan melalui pendekatan algoritmis. Beberapa algoritma yang relevan dengan analisis teori permainan seperti algoritma *Dijkstra* (*A-star*) dan *Hamiltonian Cycle*.

Dalam permainan ini, *reward* ular (agen) dalam memperoleh buah berbanding terbalik dengan risiko bertambahnya panjang tubuh yang dapat menyebabkan kekalahan. Setiap kali pemain memperoleh keuntungan (buah), tantangan juga meningkat, sehingga menghasilkan keseimbangan antara risiko dan *reward*, sebuah konsep yang umum dalam teori permainan. Teori permainan sering kali melibatkan manajemen risiko, di mana pemain harus memilih antara strategi berisiko tinggi dengan *reward* besar atau strategi konservatif dengan hasil yang lebih terprediksi. Dalam *Snake*, pemain yang terus-menerus menggunakan strategi agresif untuk langsung mendapatkan buah mungkin lebih cepat gagal, sementara pemain yang lebih berhati-hati dapat bertahan lebih lama dengan strategi yang lebih defensif, seperti menjaga ruang gerak yang luas sebelum mengincar buah (Myerson, 1991).

2.5 *Deep Learning*

Deep Learning (DL) adalah teknik yang menggunakan jaringan saraf tiruan berlapis-lapis (*deep neural networks*) untuk menangani masalah non-linear yang kompleks. Pada kasus gim seperti *Snake*, RL digunakan agar agen (dalam hal ini, pemain *Snake*) belajar bagaimana bergerak dalam grid untuk memaksimalkan *reward* biasanya dengan makan makanan dan menghindari benturan dengan tubuh atau dinding. Teori-teori utama yang ada pada *Deep Reinforcement Learning* (DRL) diantaranya adalah *Deep-Q-Network* (*DQN*), *Experience Replay*, *Target Network*, *Policy Gradient Method*.

2. 5. 1. *Deep-Q-Network (DQN)*

Banyak aplikasi dari *Reinforcement Learning* seperti mobil yang dapat menyetir sendiri, Atari, konsumen marketing, pelayanan kesehatan, edukasi dan lainnya memiliki banyak *state* (s) dan atau aksi (a). Bahkan ada bisa saja ada beberapa state yang tidak dijumpai saat model training dijalankan. Hal ini membutuhkan model representatif seperti *reward*, nilai dari *state-action* (Q), nilai dari state (V), dan peraturan yang dapat memunculkan beberapa state dan aksi. Agen mempelajari VFA (*Value Function Approximation*) dengan memperbarui bobot setiap langkah waktu berdasarkan *Temporal Difference* (TD) return saat ini. Setelah pembaruan, TD return tersebut dibuang, dan TD return baru digunakan untuk pembaruan bobot berikutnya. Metode ini tidak memaksimalkan penggunaan data yang telah diambil untuk menyesuaikan fungsi.

Istilah "*Deep Q-Learning*" digunakan karena jaringan saraf biasanya diimplementasikan dengan deep neural networks (DNNs) dalam banyak aplikasi. TD (*Temporal Difference*) adalah metode dalam *Reinforcement Learning* yang digunakan untuk memperkirakan nilai (*value*) dari suatu keadaan atau aksi berdasarkan perbedaan temporal, yaitu perbedaan antara nilai yang diprediksi saat ini dan nilai yang sebenarnya diterima di langkah berikutnya. TD menggabungkan aspek dari *Monte Carlo* dan *dynamic programming*.

Inti dari metode TD adalah TD error, yang mengukur kesalahan prediksi antara nilai saat ini dan estimasi nilai berikutnya. Pembaruan dilakukan secara bertahap setiap kali agen mengambil tindakan dan menerima umpan balik (*reward*). *TD learning* sering digunakan dalam algoritma seperti *Q-Learning* dan SARSA.

Berikut adalah contoh sederhana dari Update Temporal Difference:

$$V(s) = V(s) + \alpha[r + \gamma V(s') - V(s)] \dots \dots \dots (2.2)$$

Persamaan 2.2 merupakan rumus pembaruan nilai keadaan dalam pembelajaran penguatan (reinforcement learning). Dalam persamaan ini, $V(s)$ merepresentasikan nilai dari suatu keadaan s , sedangkan α adalah laju pembelajaran yang menentukan seberapa besar perubahan nilai berdasarkan pengalaman baru. Nilai r merupakan *reward* atau imbalan yang diterima saat berpindah dari keadaan s ke keadaan s' . Selanjutnya, γ / *gamma* adalah faktor diskon yang menunjukkan seberapa besar pengaruh nilai masa depan terhadap nilai saat ini. Terakhir, $V(s')$ adalah nilai prediksi untuk keadaan berikutnya s' .

Dengan menggunakan pendekatan ini, agen belajar dari pengalaman tanpa perlu menunggu episode selesai, berbeda dengan metode *Monte Carlo* yang memerlukan penyelesaian episode penuh untuk memperbarui nilai (Kuang, 2021).

2. 5. 2. Experience Replay

Pada metode Experience Replay dalam Reinforcement Learning, agen menyimpan data pengalaman interaksi dengan lingkungan dalam bentuk tuple (s, a, r, s') , yang masing-masing merepresentasikan keadaan saat ini (*state*), aksi yang diambil (*action*), reward yang diterima (*reward*), dan keadaan berikutnya (*next state*). Data ini disimpan ke dalam replay buffer, yaitu struktur data khusus yang berfungsi sebagai memori jangka pendek agen. Dengan menyimpan dan mendaur ulang pengalaman ini, algoritma dapat melakukan pembaruan nilai Q secara lebih stabil dan efisien. Pengambilan sampel secara acak dari *buffer* ini bertujuan untuk mengurangi korelasi antar data dan memungkinkan pemanfaatan ulang informasi berharga dari interaksi sebelumnya. Hal ini menjadikan *Experience Replay* sebagai teknik penting untuk meningkatkan efisiensi pembelajaran, terutama dalam lingkungan yang sangat dinamis seperti permainan Snake.

Pengambilan sampel dari sebagian kecil tuple untuk melatih jaringan saraf disebut *experience replay*. Selain mengurangi korelasi, *experience replay* memungkinkan kita belajar lebih banyak dari satu set tuple berkali-kali, dan secara umum memaksimalkan penggunaan pengalaman.

Untuk melakukan *experience replay*, kita secara acak mengambil sampel *batch* dari tuple pengalaman, menghitung nilai target, dan kemudian menggunakan *Stochastic Gradient Descent* (SGD) untuk memperbarui bobot jaringan saraf. Metode pelatihan jaringan saraf umum dapat digunakan untuk tujuan ini (Kuang, 2021).

2. 5. 3. Policy Gradient Method

Metode *policy gradient* mempelajari kebijakan secara langsung, sementara metode seperti TD, SARSA, dan *Q-Learning* adalah metode berbasis nilai, yang berfokus pada evaluasi dan pembaruan nilai dari tindakan atau keadaan tertentu. Teorema *policy gradient* adalah konsep penting dalam metode *policy gradient*, yang akan dijelaskan dengan bukti terperinci. Sebelum itu, dijelaskan contoh sederhana berupa MDP satu langkah, di mana proses dimulai dari keadaan tertentu dan selesai setelah satu langkah dengan pemberian *reward* $r = R_{s,a}$.

Fungsi objektifnya adalah:

$$J(\theta) = E_{\pi_{\theta}}[r] = \sum_{s \in S} d(s) \sum_{a \in A} \pi_{\theta}(s, a) R_{s,a} \dots \dots \dots (2.3)$$

Persamaan 2.3 ini menggambarkan cara menghitung ekspektasi *reward* total yang diharapkan (dinyatakan sebagai $J(\theta)$) dengan menjalankan kebijakan tertentu π_{θ} . Ekspektasi ini dihitung dengan menjumlahkan *reward* dari setiap pasangan keadaan-tindakan (s, a) , dikalikan probabilitas kebijakan untuk memilih tindakan tertentu pada suatu keadaan $(\pi_{\theta}(s, a))$, dan frekuensi terjadinya $d(s)$. Teorema *Policy gradient* pada dasarnya mencoba untuk

mengoptimalkan kebijakan dengan mengubah parameter θ untuk memaksimalkan *reward* $J(\theta)$ yang akan dihasilkan (Kuang, 2021).

2.6 Penelitian Terdahulu

Berdasarkan literatur yang digunakan dalam penulisan proposal ini, dapat disimpulkan bahwa implementasi reinforcement learning dapat membantu sistem dalam mengambil keputusan secara adaptif berdasarkan pengalaman, sehingga meningkatkan efektivitas dalam proses pengambilan keputusan. Tabel 2.1 ini adalah acuan yang memiliki keterkaitan dengan penelitian yang akan dilakukan.

Tabel 2.1 Penelitian Terdahulu

No	Nama Penulis dan Tahun Publikasi	Hasil
1	(Bakar, dkk., 2023)	<p>Judul: <i>Fusion Sparse And Shaping Reward Function In Soft Actor-Critic Deep Reinforcement Learning For Mobile Robot Navigation</i></p> <p>Metode: <i>Soft Actor Critic</i></p> <p>Hasil: Saat ini, kemajuan dalam robot otonom sangat dipengaruhi oleh perkembangan teknologi terbaru. <i>Deep Reinforcement Learning</i> (DRL) memungkinkan sistem beroperasi secara otomatis, di mana robot mempelajari gerakan berdasarkan interaksi dengan lingkungan. Salah satu pendekatan DRL terbaru adalah <i>Soft Actor Critic</i> (SAC) yang mampu mengontrol tindakan berkelanjutan untuk menghasilkan gerakan yang lebih akurat. Meskipun SAC tangguh terhadap ketidakpastian, terdapat kelemahan dalam proses</p>

No	Nama Penulis dan Tahun Publikasi	Hasil
		<p>eksplorasi yang mempengaruhi kecepatan pembelajaran. Penelitian ini mengusulkan penggunaan fungsi ganjaran berbasis ganjaran jarang dan terstruktur dalam metode SAC untuk meningkatkan efektivitas pembelajaran robot. Hasil eksperimen menunjukkan bahwa kombinasi ini berhasil menavigasi robot ke sasaran dengan meningkatkan akurasi hingga rata-rata kesalahan 4,99%.</p>
2	(Ridho dkk., 2024)	<p>Judul: Penerapan <i>Deep Reinforcement Learning</i> dalam <i>Business Intelligence</i></p> <p>Metode: <i>Deep Neural Network</i></p> <p>Hasil: <i>Business Intelligence</i> (BI) adalah kombinasi alat seperti gudang data, OLAP, dan dasbor untuk analisis data mendalam. Gudang data mengumpulkan data akurat dari berbagai sumber, sementara OLAP memungkinkan analisis multidimensi secara real-time. <i>Deep Reinforcement Learning</i> (DRL) diidentifikasi sebagai teknik yang menjanjikan untuk mengatasi tantangan pengambilan keputusan berurutan dengan ketidakpastian. <i>Deep neural network</i> (DNN) juga digunakan dalam BI untuk prediksi yang lebih baik. Temuan menunjukkan bahwa <i>Double Deep Q Learning</i> dapat meningkatkan BI</p>

No	Nama Penulis dan Tahun Publikasi	Hasil
		dengan mengoptimalkan sumber daya dan mengurangi waktu kalkulasi dalam masalah skala besar serta memodelkan pola strategi lawan dalam sistem multi-agen.
3	(Afriyadi & Utomo, 2023)	<p>Judul: <i>Towards Human-Level Safe Reinforcement Learning in Atari Library</i></p> <p>Metode: <i>Double Deep Q Learning</i></p> <p>Hasil: Penelitian ini mengusulkan pendekatan baru untuk mengintegrasikan pembatasan keselamatan yang dibuat secara manual dalam algoritme <i>Double Deep Q-Network (DDQN)</i>. Tujuannya adalah agar agen RL dapat belajar untuk bertindak dengan aman tanpa mengorbankan kinerja. Penelitian ini menggunakan pustaka Atari, khususnya gim <i>Super Mario Bros</i>, sebagai contoh lingkungan yang kompleks untuk melatih agen RL. Hasil penelitian menunjukkan bahwa metode yang diusulkan, termasuk pembatasan keselamatan, mampu mengungguli algoritme RL konvensional dalam hal keamanan dan pencapaian hasil yang optimal.</p>
4	(Taqwa dkk., 2023)	<p>Judul: <i>Designing A WSNs-based Smart Home Monitoring System through Deep Reinforcement Learning</i></p> <p>Metode: <i>Deep Reinforcement Learning</i></p>

No	Nama Penulis dan Tahun Publikasi	Hasil
		<p>Hasil: pengembangan sistem pemantauan rumah pintar berbasis <i>wireless sensor networks</i> (WSNs) dan <i>deep Reinforcement Learning</i> (DRL) untuk memantau parameter lingkungan seperti suhu, kelembapan, dan kadar CO₂. Sistem ini dirancang untuk memberikan rekomendasi apakah lingkungan rumah dalam kondisi "nyaman" atau "tidak nyaman." Proses eksperimen dilakukan melalui validasi akurasi sensor WSNs, pemilihan model DRL terbaik, dan pengujian sistem pemantauan. Hasil pengujian menunjukkan bahwa sistem ini memiliki tingkat akurasi tinggi (95.52% untuk DRL, 98.52% untuk WSNs) dan biaya rendah, serta mudah diimplementasikan. Sistem ini juga dilengkapi aplikasi berbasis Android untuk memudahkan pengguna dalam memantau kondisi rumah secara <i>real-time</i>.</p>
5	(Hidayat dkk., 2023)	<p>Judul: <i>Modified Q-Learning Algorithm for Mobile Robot Real-Time Path Planning using Reduced States</i></p> <p>Metode: <i>Q-Learning</i></p> <p>Hasil: Perencanaan jalur adalah algoritme penting dalam robot otonom, termasuk robot pertanian. Salah satu metode <i>Reinforcement Learning</i> yang digunakan untuk perencanaan jalur robot adalah algoritme <i>Q-Learning</i>. Namun, metode <i>Q-Learning</i> konvensional membutuhkan biaya</p>

No	Nama Penulis dan Tahun Publikasi	Hasil
		komputasi tinggi karena mengeksplorasi semua kemungkinan keadaan robot untuk menemukan jalur optimal. Studi ini memodifikasi algoritme <i>Q-Learning</i> dengan menghilangkan area yang tidak dapat dilalui sehingga area tersebut tidak dihitung sebagai grid yang harus diproses. Algoritme <i>Q-Learning</i> yang dimodifikasi ini diuji pada robot otonom di Agribusiness and Technology Park (ATP), IPB. Hasil simulasi menunjukkan bahwa modifikasi ini mengurangi biaya komputasi sebesar 50,71%, dengan waktu rata-rata 25,74 detik dibandingkan dengan 50,75 detik pada metode <i>Q-Learning</i> asli. Meskipun begitu, kedua metode menghasilkan jumlah keadaan yang sama untuk jalur optimal, yaitu 56 keadaan. Algoritme <i>Q-Learning</i> yang dimodifikasi juga dapat menemukan jalur dengan nilai parameter learning rate minimal 0,2 ketika faktor diskon bernilai 0,9.
6	(Zhu & Zhang, 2021)	<p>Judul: <i>Deep Reinforcement Learning Based Mobile Robot Navigation: A Review</i></p> <p>Metode: <i>Deep Q Network</i></p> <p>Hasil: Aplikasi pelaporan dan pelacakan kejahatan berhasil dibuat. Masyarakat dapat melaporkan pelaku kejahatan secara <i>realtime</i> melalui <i>smartphone</i> android. Kejahatan akan cepat ditangani menggunakan konsep <i>geofencing</i>. Di dalam aplikasi ini memiliki 2 jenis <i>user</i> yaitu</p>

No	Nama Penulis dan Tahun Publikasi	Hasil
		masyarakat dan polisi, dengan adanya fitur <i>near me</i> masyarakat dan polisi dapat melihat apa yang terjadi di sekitarnya. Masyarakat harus <i>login</i> terlebih dahulu untuk menggunakan fitur <i>panic button</i> .
7	(Alrahma dkk., 2024)	<p>Judul: <i>Application of Q-Learning Method for Disaster Evacuation Route Design Case Study: Digital Center Building UNNES</i></p> <p>Metode: <i>Q-Learning</i></p> <p>Hasil: Penelitian ini menerapkan <i>Q-Learning</i>, sebuah teknik pembelajaran penguatan, untuk merancang rute evakuasi secara efisien di lantai pertama Gedung Digital Center di UNNES, yang menghasilkan jalur optimal dengan akurasi tinggi dan efisiensi komputasi yang baik, bahkan ketika ada hambatan tambahan.</p>
8	(Changbao dkk., 2024)	<p>Judul: <i>Research on Game Character Behavior Decision-Making System Based on Deep Reinforcement Learning</i></p> <p>Metode: <i>Deep Reinforcement Learning</i></p> <p>Hasil: Makalah ini mengeksplorasi penggunaan <i>Deep Reinforcement Learning</i> (DRL) dalam meningkatkan kemampuan pengambilan keputusan karakter gim. Metode tradisional seperti Pohon Perilaku dan Mesin Keadaan Terbatas memiliki keterbatasan dalam adaptabilitas dan kinerja karena sifatnya yang</p>

No	Nama Penulis dan Tahun Publikasi	Hasil
		berbasis aturan. Studi ini menyajikan model DRL, khususnya menggunakan <i>Deep Q-Network (DQN)</i> , yang menunjukkan peningkatan kualitas pengambilan keputusan dan kecepatan respons untuk karakter gim. Penelitian ini membahas penerapan DRL dalam lingkungan gim yang kompleks, menekankan keunggulannya seperti strategi perilaku adaptif dan pengalaman bermain gim yang lebih baik. Meskipun memiliki potensi, DRL menghadapi tantangan seperti waktu pelatihan yang panjang dan interpretabilitas model yang buruk

BAB 3

METODOLOGI PENELITIAN

Pada bab ini dijelaskan metodologi yang digunakan dalam penelitian ini. Metodologi mencakup pendekatan dan langkah-langkah yang sistematis untuk mencapai tujuan penelitian, khususnya dalam konteks penerapan *Reinforcement Learning*.

Pengerjaan penelitian ini meliputi garis besar penelitian, diagram alir proses, prosedur penelitian, serta rencana jadwal pelaksanaan. Seluruh tahapan tersebut akan digunakan dalam **“IMPLEMENTASI *ARTIFICIAL INTELLIGENCE* PADA PERMAINAN SNAKE DENGAN METODE *REINFORCEMENT LEARNING*”**.

3.1 Gambaran Besar Penelitian

Gambaran besar penelitian ini adalah memberikan panduan umum mengenai alur penelitian dari awal hingga akhir. Pengerjaan "Implementasi *Artificial intelligence* pada Permainan *Snake* dengan Metode *Reinforcement Learning*" dibagi menjadi 4 tahap.

Tahap pertama adalah pengembangan lingkungan simulasi permainan *Snake* yang akan digunakan sebagai platform pelatihan untuk agen AI. Pada tahap ini, dilakukan pemodelan permainan *Snake* dengan mempertimbangkan semua variabel lingkungan, seperti ukuran papan permainan, posisi makanan, dan kondisi-kondisi permainan yang memungkinkan. Simulasi ini dirancang agar agen dapat berinteraksi dengan lingkungan secara efektif untuk mempelajari tindakan yang menghasilkan *reward* optimal.

Tahap kedua adalah penerapan algoritma *Reinforcement Learning*, seperti *Q-Learning* dan *Deep Q-Network (DQN)*, pada lingkungan simulasi tersebut. Agen AI dilatih dengan berbagai iterasi permainan, di mana setiap tindakan yang diambil menghasilkan *reward* atau penalti. Pada tahap ini, algoritma *Q-Learning* akan menggunakan tabel nilai (Q-table) untuk menyimpan *reward* potensial dari setiap

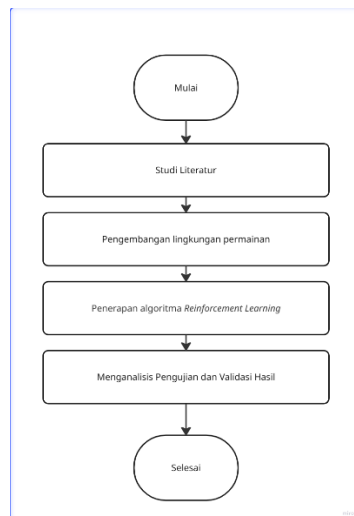
tindakan, sementara metode *DQN* akan memanfaatkan jaringan saraf tiruan untuk mengestimasi nilai *Q* dalam ruang tindakan yang lebih kompleks.

Tahap ketiga adalah analisis kinerja agen AI yang dilatih menggunakan algoritma-algoritma tersebut. Setelah agen selesai dilatih, dilakukan evaluasi terhadap kemampuan agen dalam memainkan permainan *Snake*, meliputi seberapa baik agen dapat menghindari tabrakan, memakan makanan, dan memaksimalkan skor. Hasil evaluasi ini akan menunjukkan efektivitas algoritma *Reinforcement Learning* yang digunakan.

Tahap keempat adalah pengujian dan validasi hasil. Pada tahap ini, dilakukan analisis konsistensi kinerja agen dan perbandingan antara hasil pelatihan menggunakan *Q-Learning* dan *DQN*. Diperiksa apakah hasil yang diperoleh konsisten dalam berbagai kondisi lingkungan permainan yang berbeda dan apakah agen menunjukkan kemampuan adaptasi yang baik terhadap perubahan. Hasil dari tahap ini akan menjadi dasar untuk menarik kesimpulan mengenai efektivitas metode *Reinforcement Learning* dalam permainan *Snake* dan potensi pengembangannya untuk aplikasi lain.

3.2 Diagram Alir Penelitian

Adapun diagram alir dari penelitian ini adalah sebagai berikut:



Gambar 3. 1 Diagram Alir Penelitian

Pada Gambar 3.1 ditunjukkan diagram alir dari penelitian ini. Terdapat 4 tahapan, yaitu Pengembangan lingkungan simulasi, Algoritma *Reinforcement Learning*, Analisis kinerja agen AI, dan pengujian dan validasi hasil.

3.3 Prosedur Penelitian

Adapun penjelasan lebih lengkap terkait tahapan-tahapan yang ada pada sub bab diagram alir penelitian (Gambar 3.1) akan dijelaskan pada sub bab prosedur penelitian ini. Berikut penjelasan tahap-tahap yang dilakukan untuk menyelesaikan penelitian implementasi *artificial intelligence* pada permainan *snake* dengan metode *Reinforcement Learning*.

3.3.1 Studi Literatur

Studi literatur ini terdapat pada tahap pertama untuk mengumpulkan literasi yang sesuai dengan topik penelitian dalam implementasi *artificial intelligence* pada permainan *snake* dengan metode *Reinforcement Learning*. Beberapa literatur yang akan digunakan adalah tentang metode Double Q Network, penjelasan mengenai agen AI.

Adapun hasil dari tahap studi literatur adalah didapatkan penjelasan terkait metode, dan didapatkan informasi terkait dengan metode yang tepat serta kelebihan dari metode tersebut dalam proses pengembangan sistem model AI dengan *Reinforcement Learning*. Kemudian memberikan gambaran terkait tahapan metodologi yang harus dilalui berdasarkan metode yang digunakan.

3.3.2 Pengembangan Lingkungan Simulasi

Dalam tahap pengembangan lingkungan simulasi, kami akan membuat model permainan *Snake* yang komprehensif dengan mempertimbangkan berbagai variabel penting seperti ukuran papan permainan, posisi makanan, dan kondisi permainan yang beragam. Simulasi ini dirancang untuk memungkinkan agen kecerdasan buatan (AI)

berinteraksi secara efektif dengan lingkungan, sehingga dapat mempelajari tindakan yang menghasilkan *reward* optimal. Dengan lingkungan simulasi yang realistis dan dinamis, kami bertujuan untuk menciptakan platform pelatihan yang ideal bagi agen AI untuk mengembangkan strategi bermain yang efisien dan adaptif.

Agar simulasi ini dapat mendukung proses pembelajaran agen AI dengan maksimal, kami akan memastikan setiap elemen permainan *Snake* disimulasikan secara akurat. Ini termasuk penyesuaian tingkat kesulitan permainan dan penyediaan umpan balik yang tepat untuk setiap tindakan yang dilakukan oleh agen. Selain itu, simulasi ini akan dioptimalkan untuk memungkinkan iterasi yang cepat, memungkinkan agen untuk belajar dari pengalaman dengan cepat dan efektif.

3.3.3 Penerapan Algoritma *Reinforcement Learning*

Pada tahap penerapan algoritma dalam penelitian "Implementasi *Artificial intelligence* pada Permainan *Snake* dengan Metode *Reinforcement Learning*", beberapa langkah penting dilakukan untuk memastikan agen AI dapat belajar dan beradaptasi dengan lingkungan permainan secara efektif.

Algoritma *Reinforcement Learning* yang dipilih untuk diterapkan adalah *Q-Learning* dan *Deep Q-Network (DQN)*. *Q-Learning* menggunakan tabel nilai (Q-table) untuk menyimpan *reward* potensial dari setiap tindakan, sedangkan *DQN* memanfaatkan jaringan saraf tiruan untuk mengestimasi nilai Q dalam ruang tindakan yang lebih kompleks. Algoritma tersebut diterapkan pada lingkungan simulasi yang telah dikembangkan. Proses ini melibatkan pengkodean algoritma dan integrasinya dengan simulasi permainan *Snake*, sehingga agen dapat mulai belajar dari interaksi dengan lingkungan. Agen AI dilatih melalui berbagai iterasi permainan. Setiap tindakan yang diambil oleh agen menghasilkan *reward* atau penalti, yang digunakan untuk memperbarui strategi agen. Proses ini memungkinkan agen untuk belajar dari pengalaman dan meningkatkan kemampuannya dalam bermain *Snake*. Selama pelatihan, dilakukan pengujian untuk memantau kinerja agen dan menyesuaikan

pengaturan parameter seperti jumlah episode pelatihan, nilai epsilon sebagai tingkat eksplorasi, *learning rate*, *discount rate*. Setelah agen siap, dilakukan inisialisasi terhadap lingkungan permainan Snake, yang meliputi penempatan posisi awal ular, makanan, serta arah gerak awal.

Setelah permainan dimulai, agen akan mengamati kondisi lingkungan saat ini (state) yang berisi informasi seperti arah gerak ular, posisi makanan relatif terhadap kepala ular, serta potensi tabrakan dengan dinding atau tubuhnya sendiri. Berdasarkan kondisi tersebut, agen akan menentukan aksi selanjutnya dengan pendekatan epsilon-greedy. Jika nilai acak lebih kecil dari epsilon, maka agen memilih aksi secara acak (eksplorasi), sedangkan jika lebih besar, agen memilih aksi berdasarkan prediksi terbaik dari model (eksploitasi).

Aksi yang diambil akan dijalankan dalam permainan dan menghasilkan reward serta perubahan lingkungan. Pengalaman yang terdiri dari state, action, reward, next state, dan kondisi selesai atau tidak akan langsung digunakan untuk pelatihan memori jangka pendek (*short-term memory*), dan juga disimpan dalam replay buffer untuk keperluan pelatihan jangka panjang (*long-term memory*). Ketika permainan belum selesai, agen akan kembali mengamati state baru dan melanjutkan proses pengambilan keputusan. Namun jika permainan berakhir, misalnya karena agen menabrak dinding atau tubuhnya sendiri, maka sistem akan melakukan pelatihan memori jangka panjang menggunakan batch sampel dari *replay buffer*.

Setiap akhir episode, agen juga melakukan evaluasi performa dan menyimpan model terbaik jika terjadi peningkatan *score*. Proses ini diulang hingga jumlah episode yang ditentukan terpenuhi. Setelah pelatihan selesai, model disimpan untuk digunakan kembali pada tahap pengujian atau implementasi lanjutan. Skema ini dirancang untuk memastikan bahwa agen mampu belajar secara bertahap dari interaksi dengan lingkungan, dengan tujuan akhir yaitu menghasilkan agen yang optimal, dan mampu bertahan hidup dalam permainan sembari memaksimalkan *reward*.

3.3.4 Menganalisis Pengujian dan Validasi Hasil

Setelah model agen AI dilatih menggunakan algoritma Deep Q-Network (DQN) dalam permainan Snake, tahap berikutnya adalah melakukan pengujian guna mengevaluasi performa agen yang telah dilatih. Pengujian ini dilakukan dengan membiarkan agen bermain dalam lingkungan permainan yang sama, tanpa adanya modifikasi terhadap parameter atau aturan permainan. Tujuannya adalah untuk menilai seberapa efektif agen dalam mengambil keputusan dan bertahan hidup berdasarkan pembelajaran yang telah diperoleh selama pelatihan.

Selama proses pengujian, beberapa metrik penting digunakan sebagai indikator keberhasilan, di antaranya adalah jumlah skor yang diperoleh, lama waktu agen bertahan hidup (jumlah langkah), serta konsistensi dalam pengambilan keputusan. Pengujian ini juga memberikan gambaran apakah model mampu menggeneralisasi strategi yang efektif terhadap berbagai situasi permainan atau hanya sekadar mengingat pola yang sering muncul selama pelatihan.

Selain pengujian, dilakukan pula validasi terhadap hasil pelatihan untuk memastikan bahwa model tidak mengalami overfitting dan bahwa strategi yang dikembangkan oleh agen AI memang berasal dari pembelajaran yang bermakna, bukan kebetulan atau eksploitasi dari kondisi permainan yang sempit. Validasi dilakukan dengan mengulangi permainan beberapa kali menggunakan seed acak yang berbeda, serta membandingkan performa agen dengan baseline atau model acak.

Hasil pengujian dan validasi menunjukkan bahwa agen yang dilatih dengan DQN menunjukkan peningkatan skor rata-rata dan kemampuan bertahan hidup yang lebih tinggi dibandingkan dengan agen acak. Hal ini menunjukkan bahwa penggunaan algoritma Reinforcement Learning dalam permainan Snake memberikan kontribusi signifikan dalam membentuk strategi yang adaptif dan efisien. Dari sini dapat disimpulkan bahwa agen AI telah

berhasil belajar dari lingkungannya dan dapat mengambil keputusan yang mengarah pada perolehan reward yang lebih besar.

3.4 Jadwal Penelitian

Adapun susunan rencana penelitian yang dilakukan oleh penulis berdasarkan metodologi yang telah dijabarkan sebelumnya. Susunan penelitian dapat dilihat pada Tabel 3.1 sebagai berikut.

Tabel 3.1 Rencana Penelitian

Tahap	Bulan											
	September				Oktober				November			
	1	2	3	4	1	2	3	4	1	2	3	4
Melakukan studi literasi												
Melakukan simulasi pengembangan lingkungan agen												
Implementasi algoritma												
Melakukan pelatihan pada agen												
Analisis performa agen												
Melakukan testing dan validasi												

Hasil Analisa pembelajaran agen												
Penulisan laporan												

DAFTAR PUSTAKA

- Alpaydin, E., 2020. *Introduction to Machine Learning*. 4th ed. Cambridge: MIT Press.
- Bostrom, N., 2014. *Superintelligence : paths, dangers, strategies*. 1st ed. Oxford: Oxford : Oxford University Pressv.
- Crespo, J. & Wichert, A., 2020. *Reinforcement Learning* applied to games. *SN Applied Sciences*, Volume 2, p. 824.
- François-Lavet, V. et al., 2018. An Introduction to Deep. *Foundations and Trends in Machine Learning*, 11(3–4), p. 219–354.
- Goodfellow, I., Bengio, Y. & Courville, A., 2016. *Deep Learning*. Massachusetts: MIT Press.
- Harrington, P., 2012. *Machine Learning in Action*v. 1 ed. New York: Manning Publications.
- Kuang, W., 2021. *Fundamentals of Reinforcement Learning*. Edinburg: University of Texas Rio Grande Valley.
- Li, Y., 2018. *Deep Reinforcement Learning. Clinical Orthopaedics and Related Research*, Volume abs/1810.06339.
- Mitchell, T. M., 1997. *Machine Learning*. New York: McGraw-Hill Education.
- Mnih, V. et al., 2015. *Human-level control through deep Reinforcement Learning*. s.l.:s.n.
- Myerson, R. B., 1991. *Game Theory: Analysis of Conflict*. Cambridge: Harvard University Press.
- Russel, S. & Norvig, P., 2020. *Artificial Intelligence A Modern Approach Fourth Edition*. London: Pearson.

- Silver, D. et al., 2016. Mastering the game of Go with deep neural networks and tree search. *Nature*, Volume 529, p. 484–489.
- Sutton, R. S. & Barto, A. G., 2018. *Reinforcement Learning: An Introduction*. 2nd ed. Cambridge: MIT Press.
- Taqwa, A., Isa, I. G. T. & Ariyanti, I., 2023. Designing A WSNs-based Smart Home Monitoring System through. *Rekayasa Sistem dan Teknologi Informasi*, pp. 1224-1232.
- Tegmark, M., 2017. *Life 3.0 : being human in the age of artificial intelligence*. New York: Alfred A. Knopf.